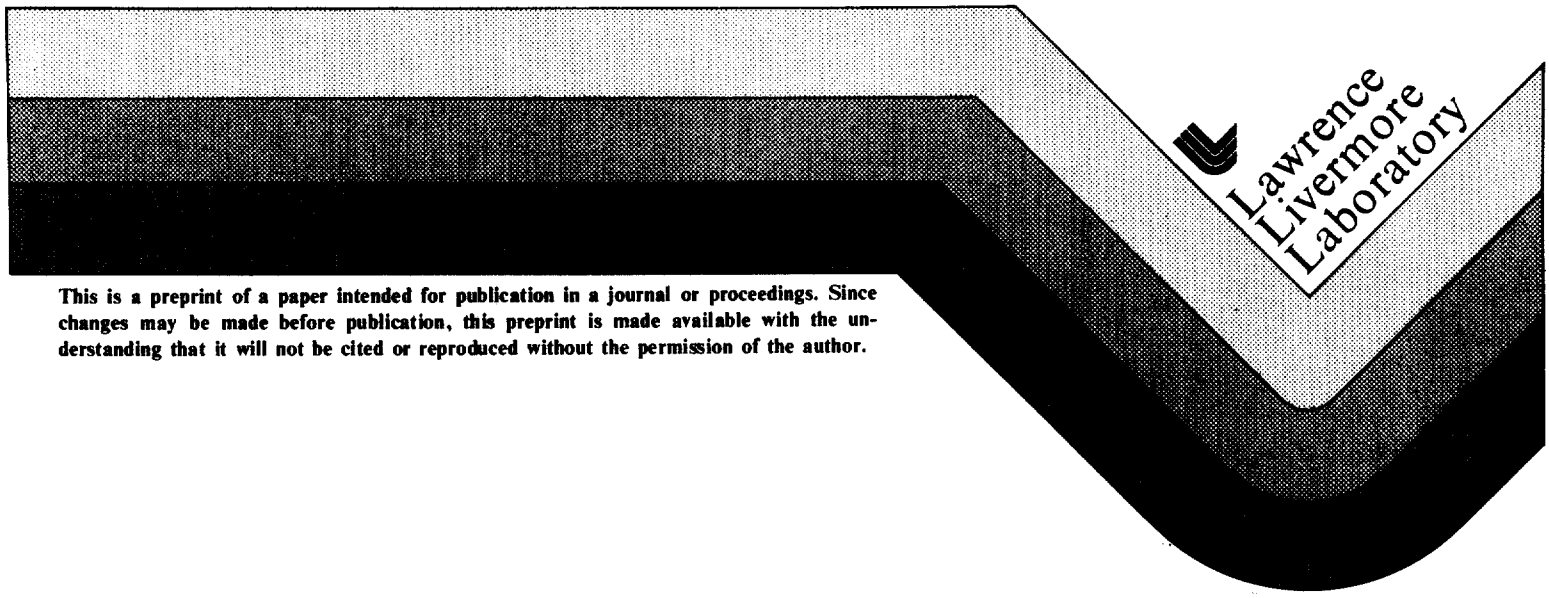
UCRL-83045
PREPRINT

# FLUID STRUCTURE COUPLING ALGORITHM

William H. McMaster

This paper was prepared for submittal to
The 1980 PVP Technology Conference in
San Francisco
August 13-15, 1980

January 1980



# FLUID STRUCTURE COUPLING ALGORITHM

W. H. McMaster
E. Y. Gong
C. S. Landram
D. F. Quinones
University of California, Lawrence Livermore Laboratory
P.O. Box 808, Livermore, California, 94550

*ABSTRACT*

*A fluid-structure-interaction algorithm has been developed and incorporated into the two dimensional code PELE-IC. This code combines an Eulerian incompressible fluid algorithm with a Lagrangian finite element shell algorithm and incorporates the treatment of complex free surfaces. The fluid structure, and coupling algorithms have been verified by the calculation of solved problems from the literature and from air and steam blowdown experiments. The code has been used to calculate loads and structural response from air blowdown and the oscillatory condensation of steam bubbles in water suppression pools typical of boiling water reactors. The techniques developed here have been extended to three dimensions and implemented in the computer code PELE-3D.*

## NOMENCLATURE

| | |
|---|---|
| c | Speed of sound in the fluid |
| D | Divergence |
| F | structure iterative constants |
| g | Gravitational constant |
| K | Stiffness matrix |
| $\dot{m}$ | Mass flow rate |
| M | Mass matrix |
| P | Pressure |
| p | P/ρ, ratio of pressure to density |
| Q | Generalized displacement vector |
| t | Time |
| u | Velocity |
| V | Volume |
| z | Distance |
| ε | Convergence constant |
| γ | Ratio of specific heats |
| ϕ | Over-relaxation constant |
| ρ | Mass density |
| τ | Dummy variable |
| ν | Kinematic viscosity |

### SUBSCRIPTS

| | |
|---|---|
| i | Iteration number |
| o | Refers to original volume |
| u | Refers to ullage volume |

### SUPERSCRIPTS

| | |
|---|---|
| n | Time level |

## INTRODUCTION

We have developed a fluid-structure-interaction algorithm for the analysis of the dynamic response of coupled fluid structure systems. The method is incorporated into a two-dimensional semi-implicit Eulerian hydrodynamics code, PELE-IC. The code is quasi-two phase since we can couple to either a one-dimensional or a lumped parameter description of compressible gases. The code is written in both plane and cylindrical coordinates in order to handle a variety of geometrical configurations. The coupling algorithm is general in nature and can accommodate a wide variety of structural shapes. It is capable of following large interface motions through the calculational grid. By the use of a variable time step we are able to accommodate varying flow conditions and maintain computational stability. The fluid, structure, and coupling algorithms have been verified by calculations of solved problems from the literature and by comparison with air and steam blowdown experiments (1), (2).

The basic semi-implicit solution algorithm contained in the SOLA code (3) was used as a foundation for the development of the PELE-IC code. We track the movement of free surfaces using a donor cell treatment based on a combination of void fractions and interface orientation. This gives us great versatility in following fluid-gas interfaces for bubble definition and water surface motion without the use of marker particles.

The structural motion is computed by a finite element code (4) from the applied fluid pressure at the fluid structure interface. The finite element shell structure algorithm uses conventional thin-shell theory with transverse shear. The spacial discretization employs piecewise-linear interpolation functions and one-point quadrature applied to conical frustra. We use the Newmark implicit time integration method implemented as a

one step module. The fluid code then uses the structure's resultant position and velocity as boundary conditions. The fluid pressure field and the structure's response are corrected iteratively until the normal velocities of the fluid and structure are equal. This results in a strong coupling between the two algorithms.

## GENERAL DESCRIPTION OF THE SOLUTION ALGORITHMS

The underlying approach used by PELE-IC for the solution of general flow fields is the use of the semi-implicit SOLA algorithm. The basic assumption of this approach is that all flow variables within the computational grid satisfy the continuity equation for each cell, regardless of whether or not the computational cell contains a free surface or a moving structure. For incompressible fluids this means that all cells are divergence free. This assumption permits freedom of motion for all surfaces throughout the grid. Superimposed on this basic algorithm we have applied the boundary conditions for free surfaces, compressible gases, and moving structures.

In this section we will give a brief description of the solution algorithms.

### SOLA Solution Algorithm

The SOLA algorithm uses a Newton-Raphson iteration on the pressure field to solve the mass conservation equation. At each iterative step the pressure in each fluid cell is adjusted to satisfy the divergence criteria. In this algorithm the pressure is a cell centered variable and the velocity components are specified on cell sides.

The algorithm is solved by first writting the Navier-Stokes equation for the fluid velocity, u, in terms of the time level:

$$\partial u/\partial t = (-\nabla\cdot uu + g + \nu\nabla^2 u)^n - \nabla p^{n+1} \qquad (1)$$

where the superscript n indicates the time level and $p = P/\rho$ is the ratio of the pressure to the density of the fluid. The body acceleration is given by g and the kinematic viscosity is specified by the constant $\nu$. Setting $p^{n+1} = p^n + \delta p$ gives

$$u^{n+1} = [u + \delta t\ (-\nabla\cdot uu = \nabla p + g + \nu\nabla^2 u)]^n + \delta t\delta\nabla p \qquad (2)$$

Defining the term inside the brackets as $\tilde{u}$, then the equation to be solved is

$$u^{n+1} = \tilde{u} + \delta t\delta\nabla p \qquad (3)$$

where $\tilde{u}$ is found using a slightly modified form of the finite difference formulation of Hirt, et al. (3). This equation is solved iteratively where we define the divergence error, D, for each cell at the ith iteration as

$$\nabla\cdot u_i = D_i \qquad (4)$$

and $\tilde{u}$ is used as the first trial velocity to start the iteration process. The pressure increment in each cell necessary to update the velocity field is given by

$$\delta p_i = \frac{-(1+\phi)D_{i-1}}{\partial D/\partial p} \qquad (5)$$

where $\phi$ is a correction term ($0 < \phi < 1$) dependent upon adjacent cells in the direction of the sweep through the grid, and $\partial D/\partial p$ is a constant dependent only upon the cell size, the time step, and the presence of a structural boundary. We update the velocity field in each cell with the pressure increment, using

$$\delta u_i = \pm\delta p_i \delta t/\delta z \qquad (6)$$

where $\delta z$ is the cell side in the direction of u, and the sign is chosen dependent upon which side centered velocity component is being adjusted. Satisfaction of the continuity equation in any particular cell perturbs the velocity field of its neighbors. Hence, the method is applied in sweeps throughout the grid until the divergence error everywhere satisfies

$$\nabla\cdot u_i = D_i \leq \epsilon \qquad (7)$$

where $\epsilon$ is a preset convergence tolerance which should be set according to the minimum flow field of interest in the solution. The final velocity and pressure fields are then

$$u^{n+1} = \tilde{u} + \Sigma_i \delta u_i \quad \text{and} \quad p^{n+1} = p^n + \Sigma_i \delta p_i \qquad (8)$$

Since the solution procedure is a Newton-Raphson iteration, the rate of convergence is dependent upon the magnitude of $\partial D/\partial p$ which has the form

$$\frac{\partial D}{\partial p} = 2\delta t \left[\frac{F_x}{\delta x^2} + \frac{F_y}{\delta y^2}\right] \qquad (9)$$

where $F_x$ and $F_y$ are dependent upon structural interfaces coupled to the fluid cell. If there is no structure, then $F_x = F_y = 1$. From the formula for $\partial D/\partial p$ we see that convergence is accelerated by the use of large time steps and small cell sizes. However, the user is limited in his choice by the physics of the problem. In general, we require that

$$\frac{u\delta t}{\delta z} < 0.4 \qquad (10)$$

where $\delta z$ is the component $\delta x$ or $\delta y$ in the direction of the maximum velocity u.

### Thin Shell Algorithm

The finite element module uses simple shell theory with transverse shear (see Kraus (5)). The element formulation was described by Hughes and Taylor (6) for beams and plates, and was extended to axisymmetric and plane shells by Goudreau (7). (Similar results were obtained by Zienkiewicz et al. (8) at about the same time.) The element is a two-node, conical frustrum with three degrees of freedom per node. Shape functions are piecewise-linear for displacements and rotations. The shear "locking" associated with low-order interpolation is removed by one-point quadrature. Large deformation (here two to three shell thicknesses) is accounted for in an approximate way by reformulating the stiffness matrix at every time step.

The Newmark implicit time integration scheme (see Goudreau and Taylor (7)) is used at each time step to move the shell. The algorithm has the form

$$(K + 4M/(\delta t)^2)Q^{n+1} = P^{n+1} - 4M\, A^n/(\delta t)^2 \qquad (11)$$

where

$$A^n = Q^n + \dot{Q}^n \delta t + \ddot{Q}^n (\delta t)^2/4 \qquad (12)$$

Goudreau (2) gives the derivation of K and a FORTRAN listing of the one-step module.

The thin shell algorithm has been made more general by the addition of the following four features:

(1) Each element may have its own thickness.

(2) Each node can be specified to have its own separate restraints and prescribed initial displacement.

(3) The code computes the static deflection of the structure as a result of the initial loading before beginning the dynamic solution.

(4) The gas pressure in the ullage region is applied to the shell as well as fluid pressures.

Fluid Structure Interface Algorithm

This algorithm couples the fluid motion to the structure's motion within the SOLA iteration loop. Normal velocity compatibility between the structure and fluid is required where the Lagrangian shell crosses either the I-line or J-line intercept which defines the centroid of the Eulerian cell. The choice depends on the angular orientation of the structure, e.g., for angles equal or less than 45° we use the I-line coupling. The cell side coupled velocity is the one closest to the structure along the intercept line. In this manner, we maintain a smooth coupling whenever the structure crosses an Eulerian grid line. The finite element module uses the pressure field supplied by the fluid and provides the fluid code with the resultant position and velocity of the interface. Each change in the pressure field causes a different structural reponse, and each different response changes the flow field of the fluid. Therefore, the iteration proceeds until both conditions are satisfied. Within a single iteration, all Eulerian fluid zones are adjusted one by one, using the latest values available, and then all the Lagrangian shell nodes are simultaneously adjusted by the implicit time step solution.

The pressure applied to an element is determined by an interpolation along each intersecting I or J line to the neighboring full fluid cell. These interpolated values are weighted by the liquid content of the cell so that the proper pressure is applied when a free surface is in the same cell. The interpolation procedure provides a smooth pressure history whenever the structure crosses a grid line.

The solution strategy is to first set the normal fluid velocity equal to the normal structure velocity at the coupling point. The structure's normal velocity is found by an interpolation between nodal values and the intercept angle. The normal fluid velocity is found by an interpolation between all four of the cell side velocities. This determines the cell side velocity which is coupled to the structure. This first step of setting the coupled Eulerian cell velocity to satisfy the boundary conditions imposed by the structure causes the cell not to satisfy the divergence criteria; therefore, the second step is to adjust the cell pressure using the SOLA algorithm so that the cell is divergence free. This two step process is repeated each iteration until both conditions are satisfied.

The Free Surface Algorithm

Accurate free surface tracking is necessary to allow the application of velocity and pressure boundary conditions at fluid-gas interfaces. We track the free surface by a combination of void fraction and surface orientation in each cell. The void fraction provides for the conservation of mass and the surface orientation allows us to apply the proper boundary conditions and follow the flow from cell to cell.

The free surface algorithm performs four functions:

(1) Determines the surface orientation within the calculational cell based upon its fluid content and that of neighboring cells. This orientation is specified by its intercepts on two sides of the cell. Within the cell, the interface is considered to be a straight line segment. Thus, the surface is tracked by its intersection of grid lines.

(2) Applies the prescribed boundary pressure to the fluid surface. This is done by finding the appropriate cell centered pressure by an interpolation from the nearest full fluid cell to the boundary. Recent additions to the code also allow the application of a prescribed boundary velocity to the fluid surface. This option allows one to drive the surface with a moving piston. Both these options allow the boundary conditions to be a function of time.

(3) Calculates the fluid advection based on surface orientation using the donor cell method where the amount of liquid advected is determined from the contents of the upstream cell, the orientation of the surface, and the velocity of the common liquid side. This method guarantees the conservation of mass during advection.

(4) Uses velocity boundary conditions for the void sides of the cell to maintain continuity of the flow field. This assures a smooth flow when a surface crosses grid lines.

Special Features

The main application of the code to date has been to studies of the pressure suppression systems of boiling-water reactors during postulated loss-of-coolant accidents. Consequently, various special features have been added directed toward the solution of these problems. Some of these special features are described in this section.

Downcomer Pipes

Downcomer pipes are modeled by specifying the bounding grid lines as rigid. A special algorithm has been added to the code to allow this option. In this manner, pipe wall thicknesses small in comparison with a calculational cell can be correctly modeled. For vent clearing problems, the specified driving pressure is applied as a boundary condition between the grid lines defining the pipe. The code has the capability of handling up to two rigid downcomer pipes with driving pressures in this manner.

Obstacles and Baffles

Obstacles and baffles which restrict the flow can be modeled by specifying portions of grid lines as rigid boundaries. The code will then apply the boundary condition of zero normal velocity at this boundary. There is no restriction on the number of such obstacles that can be specified.

Coupling to Compressible Gas Flow

In many applications the downcomer is driven from a drywell with either variable or constant pressure. Sometimes this flow is further controlled by the use of an orifice. To provide for these situations, a flow model coupled to the fluid dynamics was developed. This model couples the bubble pressure to the drywell and current bubble volume by the equation

$$P(t) = P_u \left(\frac{V_o}{V}\right)^{\gamma} \left[1 + \frac{\gamma}{P_u V_o} \int_o^t \dot{m} V^{\gamma-1} \, d\tau\right] \qquad (13)$$

where

$\dot{m}$ = mass flow rate through the orifice as specified in Vennard (9). The formula used depends upon whether the flow is choked or unchoked.

$V_o$ = original downcomer volume from the orifice to the water level

$V$ = current steam volume including the bubble.

$P_u$ = initial ullage pressure.

The time of integration, t, covers vent clearing and subsequent bubble formulation and growth.

In application we find that the mass flow is initially choked and dependent only upon the drywell pressure and density. Subsequently, during vent clearing, the flow becomes unchoked and is dependent upon both the drywell pressure and the bubble pressure. Since the bubble pressure is dependent upon the bubble growth in the pool, there is a coupling between the suppression pool and the drywell.

Variable Ullage Pressure

During a vent clearing event, the bubble growth causes a pool swelling in the confined ullage region. This compressed air region then provides an upload on the confining structure. We derive this pressure pulse from the perfect gas law using the ullage volume change as calculated from the rise of the water surface. In experiments performed at the Massachusetts Institute of Technology (10), the test configuration applied this ullage pressure to the bottom flexible plate. The code has been modified to simulate these experiments.

Collapsing Bubbles

In chugging studies of collapsing bubbles, we have applied a condensation model to provide the applied bubble pressure. This pressure is dependent upon the inflow rate of steam and the condensation rate; both of which are dependent upon the bubble volume time history. The use of the void fraction and surface orientation algorithms allow us to monitor the bubble volume accurately.

Compressibility Effects

In the mass continuity equation, the incompressible assumption sets $\partial\rho/\partial t = 0$. We may take into account small changes in compressibility by substituting the wave equation

$$\frac{\partial \rho}{\partial t} = \frac{1}{c^2} \frac{\partial P}{\partial t} \qquad (14)$$

into the mass equation, where c is the speed of sound in the fluid. This, then, changes the specification of the divergence leading to

$$D = \nabla \cdot u + \frac{1}{c^2} \frac{\partial P}{\partial t} \qquad (15)$$

and

$$\frac{\partial D}{\partial p} = 2\delta t \left[\frac{F_x}{\delta x^2} + \frac{F_y}{\partial y^2} + \frac{1}{2c^2 \delta t^2}\right] \qquad (16)$$

which are used in the iteration and for setting the velocity boundary conditions.

SUMMARY

We have developed three new algorithms to treat free surfaces, fluid-structure boundaries, and steam condensation. The first is an air-water surface algorithm that has been used to model bubble growth and pool swell in reactor pressure suppressure systems. The second is a fluid-structure coupling algorithm that correctly couples the Lagrangian structure overlaying the Eulerian grid. The third provides the driving pressure for bubble growth and collapse dominated by steam condensation.

These algorithms have been incorporated into a three-dimensional version of the code, called PELE-3D. With this version, we are able to study nonsymmetric effects.

ACKNOWLEDGEMENTS

This work was performed under the auspices of the U. S. Department of Energy by Lawrence Livermore Laboratory under contract No. W-7405-Eng-48. The work was sponsored by the Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, and is designated Task 189:A0116 under interagency agreement DOE 40-550-75.

REFERENCES

1 McMaster, W., Norris, D., Goudreau, G., Quinones, D., Gong, E., Moran, B., and Macken, N., "Coupled Fluid-Structure Method for Pressure Suppression Analysis," NUREG/CR-0607, UCRL-52620, 1979, University of California, Lawrence Livermore Laboratory, Livermore, Calif.

2 McMaster, W., and Gong, E., "User's Manual for PELE-IC: A Computer Code for Eulerian Hydrodynamics," UCRL-52609, 1979, University of California, Lawrence Livermore Laboratory, Livermore, Calif.

3 Hirt, C. Nickols, B. and Romero, N., "SOLA -- A Numerical Solution Algorithm for Transient Fluid," LA-5852, 1975, Los Alamos Scientific Laboratory, Los Alamos, NM.

4 Goudreau, G., "A Computer Module for One Step Dynamic Response of An Axisymmetric or Plane Linear Elastic Thin Shell," UCID-17730, 1978, University of California, Lawrence Livermore Laboratory, Livermore, Calif.

5 Kraus, H., Thin Elastic Shells, John Wiley & Sons, Inc., New York, 1967.

6 Hughes, T., and Taylor, R., "Computational Methods in Applied Mechanics and Engineering," SMiRT5 Proceedings, M2/1, 1977.

7 Goudreau, G., and Taylor, R., Computational Methods in Applied Mechanics and Engineering, Vol. 2, No. 1, 1978, pp. 69-97.

8 Zienkiewicz, O., Bayer, J., Morgan, K., and Omate, E., "A Simple and Efficient Element for Axisymmetric Shells," Int. Journal of Numerical Methods in Engineering, Vol. 11, No. 10, 1977, pp. 1545.

9 Vennard, J., Elementary Fluid Mechanics, 3rd Ed., John Wiley & Sons, Inc., New York, 1954, pp. 110-111 and 301-305.

10 Anderson, W., Huber, P., and Sonin, A., "Small Scale Modeling of Hydrodynamic Forces in Pressure Suppression Systems," NUREG/CR-0003, prepared for the U.S. Nuclear Regulatory Commission, Massachusetts Institute of Technology, Dec. 1977.